UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/640,620	08/12/2003	Arra E. Avakian	10017134-1	1119

22879 7590 09/10/2008

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
VU, TUAN A	
ART UNIT	**PAPER NUMBER**
2193	

NOTIFICATION DATE	DELIVERY MODE
09/10/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
mkraft@hp.com
ipa.mail@hp.com

Office Action Summary	Application No.	Applicant(s)	
	10/640,620	AVAKIAN ET AL.	
	Examiner	Art Unit	
	TUAN A. VU	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 June 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-11 and 15-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-11, 15-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 6/17/08.

As indicated in Applicant's response, claims 1, 15, 20 have been amended. Claims 1-11, 15-20 are pending in the office action.

Specification

2. The disclosure is objected to because of the following informalities: The "Related Applications" section (Specs, pg. 1) includes U.S. applications shown only as Attorney Docket number. An update in order to provide any corresponding USPTO application/patent number would be more compliant. Appropriate correction is highly recommended.

3. The syntax shown as 'can be used the plug-in instrument objects' related to the listing of FIGUREs 9-12 at pg. 4 require some typographical correction.

Double Patenting

4. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

5. Claims 1, 20 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 15 of copending Application No. 10,640,619 (hereinafter '619) in view of Labadie, USPubN: 2003/0195959 . Although the conflicting claims are not identical, they are not patentably distinct from each other because of the following conflicts in the respective applications.

As per instant claim 1, '619 claim 15 also recites in a J2EE application server comprising transaction hierarchical parent-child transaction, comprising *first tool* selectively instrumenting top and child transaction during load process; and *second tool* selectively instrumenting top and child transaction before JVM load process;

and this is a obvious language variation of claim 1 reciting of, respectively, 'instrumenting a selected transaction by instrument hook upon execution of said selected transaction, and 'instrument hook prior to execution of selected transaction' (Note: J2EE application execution reads on JVM); '619 further recites:

'transactions in a hierarchical parent-child transaction, and instrumented wrapper objects to spawn correlators that correspond to child transactions in said hierarchy, and generating a top level correlator at a top level transaction';

and this is a obvious variant of claim 1 reciting of ' initiating top level transaction, and generating (as in spawning) correlators for identifying top level transaction upon execution of said instrumented transactions', which amount to generating plurality of correlators from all transactions identified by said parent or top level. '619 claim 15 further recites web server in response to a request transmits a cookie and the method using the cookie to generate correlator

corresponding to said top transaction; and this is a language variant of claim 1 reciting of transmitting a cookie from web server in response to a request to initiate said top level transaction, and utilizing said cookie for correlator identifying said top level.

However, '619 claim 15 does not recite 'utilizing correlators to cross-correlate a performance metric … with one or more performance metrics … one or more child transactions of said parent transaction', wherein plug-in instruments implement an interface that communicate performance metric. Labadie teaches correlators to cross-correlate performance between child and parent transaction, by way of instrument hooks implemented as plug-ins (e.g. *response time* - para 0005-0014, pg. 1-2; *plug-in* - para 0034-0035, pg. 4; Fig. 2). It would have been obvious for one skill in the art at the time the invention was made to implement '619 hooks as plug-ins and to use said correlators to cross-correlate child/parent transaction in terms of performance metrics as taught by Labadie. The motivation therefor relies on that the use of plug-ins would support dynamic code insertion and invocation without undue code translation, and that correlator usage as by Labadie and '619 is purported to inform runtime relationship between transaction entities inter-related so to yield instrumentation information, regarding which, performance metrics from instrumentation plug-ins would be a primordial interest for such information obtaining by approach like in '619.

As per instant claim 20, this claim recites installing instrument hook prior to execution and upon execution of selected transaction as in instant claim 1, using plug-ins, generating correlator for parent transaction, generating correlators for each of the transactions, to correspond top transaction and a parent transaction, to its associated transactions, and in response to initiating said top level, sending a cookie from a web server with said request and utilizing the

cookie to generate correlator for said top level. '619 claim 15 in light of the above analysis teach a variant language deemed obvious to that of instant claim 20. '619 does not recite 'performance metric corresponding to selected transaction of a plurality of parent/child transactions', and *plug-in* instruments called by hooks, *plug-ins* implementing 'an interface that communicates performance metric' but based on the teachings of Labadie, the plug-ins and the performance metric from instant claim 20 would have been an obvious feature of '619 claim 15, for the same reasons as set forth above.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-11, 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Labadie et al, USPubN: 2003/0195959 (hereinafter Labadie), and further in view of Hind et al, USPN: 7,003,565 (hereinafter Hind).

As per claim 1, Labadie discloses in a J2EE application server a method for monitoring performance of a plurality of transactions including a top level transaction and plurality of transactions relating to said top level transaction in a child parent hierarchy (e.g. Tivoli ARM … International Business Machines - para 0005-0014, pg. 1-2; para 0036, pg. 4; *event originated; event that triggered the particular event* - para 0045-0052, pg. 4-5 – Note: ARM and Tivoli

correlators reads on parent/child transaction correlators with associated measurements via API, correlation that identifying originating or triggering events/host name), comprising:

for each of selected ones of said plurality of transactions, obtaining a performance metric (e.g. *response time* - para 0005-0014, pg. 1-2) corresponding to the selected transaction by:

installing an instrument hook prior to execution of the selected transaction (e.g. FTivoli ARM … International Business Machines, *correlator data structure* - para 0005-0014, pg. 1-2; *class data structure* - para 0036, pg. 4; *event originated; event that triggered the particular ... event fields of data structure* - para 0045-0052, pg. 4-5 – Note: enabling code, format, counter specification based on event identifier, correlator, class structure reads on instrumenting prior to execution); and

instrumenting said selected transaction upon execution of the selected transaction (e.g. Fig. 4A-C; Fig. 5A-B - Note: Middleware instrumenting of live events and transaction threads reads on live hooks onto the events between selected client and server transactions, i.e. via API invoked during loaded transactions – see para 0059, pg. 5) using one or more plug-in instruments called by the instrument hook (e.g. class 310 *implements* an service provider – para 0059, pg 5; Fig. 2 – Note: service provider invoking DCS for a logging service wherein a class is instantiated to implement a logging service reads on plug-in class invoked by the instrumentation service of the DCS);

initiating said top level transaction in response to a request received from a web server (e.g. para 0032-0034, pg 3; Fig. 2 – Note: server receiving a inboud request from client DCS reads on initiating a start for a top level leading to detecting of more partner correlation or associated correlators with respect to said top level start request from DCS client);

generating correlators for identifying said top level transaction and a parent transaction, if any, upon execution of each of said instrumented transactions (e.g. para 0012-0013, pg. 2 - Note: ARM correlator reads on child/parent relationship – see para 0005, pg. 1);

utilizing said correlators to cross-correlate a performance metric corresponding to a parent transaction with one or more performance metrics corresponding to one or more child transactions of said parent transaction (e.g. Fig. 5B; *SOAP parameters, timestamp* – para 0073, pg. 7; Fig. 6A-C - Note: ARM with correlation service reads on corresponding correlator of child and that of parent),

wherein the one or more plug-in instruments implement an interface that communicates data for the performance metric (e.g. *response time* - para 0005-0014, pg. 1-2; Fig. 5B; *SOAP parameters, timestamp* – para 0073, pg. 7; Fig. 6A-C; *plug-in* - para 0034-0035, pg. 4).

Labadie specifies that the server system is a EJB server with applications (para 0026, pg. 3) involving Sun Microsystems Enterprise beans; hence has disclosed that this server system is a J2EE because of the transaction-related ARM services and instrumentation on EJB Java objects (see Fig. 6A-C).

Labadie does not explicitly disclose transmitting a cookie from said web server to said application server together with said request; and further utilizing said cookie to generate the correlator identifying said top level correlator. But client state being collected and passed (see Fig. 5A-B) over to different servers (plug-in middleware, DCS correlator service) using the instrumentation service (ARM) to record correlator as shown by Labadie (see para 0028-0032, 0034-0036) entails client runtime data/events to be passed from services to services to enable correlation of thread or partners being enumerated for a process request or analysis thereof(see

Fig. 4A-C). The use of cookie at a given machine to store client data for repeated usage – so to obviate creation of additional queries or discovery resources -- was concept used in the data collecting paradigm by Hind so that by using these record or cookie under the provision of message as to communicate with servers (see Fig. 3A; correlator – col. 8, lines 20-67) Hind's collection of correlator-type of data can support service as to improve QoS delivery or administrative policy enforcing. Hence, in the same endeavor as analyzing performance of transaction as Labadie, Hind provides in-bound and out-bound message with cookie data so to provide very specific client data for server to enforce quality control transaction using correlation information therein (see Fig. 6) thus to alleviate dependency of information interchanges from many sources of data providers (or linked servers) as cookie messaging can yield latest state of client information. Hence, in view to the multiple agent messaging as required in Labadie's correlator record passing, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide cookie messaging by Hind, i.e. using said cookie data to implement creation of the correlator to support data collection from the top level request as purported by Labadie. One skill in the art would be motivated to do so because cookie data from one sending edge server to the next would alleviate extraneous discovery resources for these data reflect the most accurate and dynamic state of the client information being passed (see Hind, col. 16, bottom to col 17 line 34) such that by utilizing this cookie approach, the servers can make use of the most up-to-date state of a client/requesting source data to fulfill the quality of transaction as approached by Labadie's instrumentation service, or to facilitate the enforcement of transaction security as endeavored by Hind's Qos paradigm.

As per claim 2, Labadie discloses the step of instrumenting said transaction comprises inserting instrumentation code in a bytecode representation of said selected transaction (byte stream – para 0072, pg. 6).

As per claims 3-6, Labadie discloses wherein said performance metric corresponds to a response time of said transaction (*response time* - para 0005-0014, pg. 1-2); wherein said instrumentation code effects generation of a start time marker upon start of execution of said selected transaction and generation of a stop time marker upon completion of execution of said selected transaction (para 0068, pg. 6); wherein said instrumentation code generates calls to an Application Response Measurement (ARM) agent to cause generation of said stop and start time markers (service 350 – Fig. 5B; para 0005-0014, pg. 1-2) utilizing said start and stop time markers to measure a response time of said selected transaction (Fig. 5A, 6A).

As per claim 7, Labadie discloses generating a record for each instrumented transaction upon completion of said instrumented transaction, said record indicating said performance metric associated with said instrumented transaction (Fig. 5A-B), a parent of said instrumented transaction, and said top level transaction (Note: for each byte stream being instrumented for a ARM code instrumentation as disclosed, the top level or correlated event being monitored reads on parent or top level transaction – see para 0045-0052, pg. 4-5).

As per claim 8, Labadie discloses transmitting said instrumented transaction record to an analysis and presentation module (e.g. *PushCorrelator, GetAllCorrelator* - Fig. 6B; *Set_Context_Data, Set_Context_Info* - Fig. 6A, B; *CorrelatorTableEntry 390* – Fig. 5B).

As per claims 9-10, Labadie discloses storing of said correlators in a thread local storage stack (e.g. Fig. 4A-C - Java Virtual Machine runtime thread with Thread counter reads on thread

stack in JVM, stack being inherent to a JVM runtime as evidenced by *Push*Correlator, *Pull*Correlator – Fig. 5B) in case of execution of said hierarchical transactions in a single thread (para 0037-0045, pg. 4); and storing said correlators in the stack based on a LIFO protocol (see Fig. 4A-C - Note: Java Virtual Machine runtime stack for threads recording with inclusion of associated correlators therein at runtime, reads on LIFO protocol of a given stack).

As per claim 11, Labadie discloses removing one correlator of the instrumented transaction's correlator from said stack upon completion of a said hierarchy of transaction associated with said correlator (*PullCorrelator* – Fig. 6B – Note: parent/child transactions being monitored – see Fig. 4A-C; Fig. 6B -- reads on hierarchy being correlated with middleware invocation).

As per claim 20, Labadie discloses a computer readable medium comprising instructions operable by a computer which when executed causes the computer to perform a method comprising:

obtaining a performance metric (e.g. *response time* - para 0005-0014, pg. 1-2) corresponding to a selected transaction of a plurality of parent-child transactions by installing an instrument hook prior to execution of the selected transaction (e.g. Tivoli ARM … International Business Machines, *correlator data structure* - para 0005-0014, pg. 1-2; *class data structure* - para 0036, pg. 4; *event originated; event that triggered the particular ... event fields of data structure* - para 0045-0052, pg. 4-5 – Note: enabling code, format, counter specification based on event identifier, correlator, class *structure* reads on instrumenting prior to execution); and

instrumenting said selected transaction upon execution of the selected transaction using one or more plug-in instruments called by the instrument hook (e.g. e.g. Fig. 4A-C; Fig. 5A-B - Note: Middleware instrumenting of live events and transaction threads reads on live hooks onto the events between selected client and server transactions, i.e. via API invoked during loaded transactions – see para 0059, pg. 5); and

generating correlators for each of said transactions, wherein each correlator identifies said top level transaction and a parent transaction, if any, corresponding to its associated transaction (e.g. para 0012-0013, pg. 2 - Note: ARM correlator reads on child/parent relationship – see para 0005, pg. 1),

wherein said top level transaction is initiated in response to a request received from a web server (para 0032-0034, pg 3; Fig. 2 – see Note in claim 1),

wherein the one or more plug-in instruments implement an interface that communicates data for the performance metric (e.g. *response time* - para 0005-0014, pg. 1-2; Fig. 5B; *SOAP parameters, timestamp* – para 0073, pg. 7; Fig. 6A-C; class 310 *implements* an service provider – para 0059, pg 5; Fig. 2).

Labadie does not explicitly disclose wherein said web server transmits a cookie to an application server and utilizing said cookie to generate said correlator for said top level transaction. But the server's transmitted cookie being used to generate additional correlators for identifying children transactions associated with top level has been addressed in claim 1.

8. Claim 15-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Labadie et al, USPubN: 2003/0195959 and Hind et al, USPN: 7,003,565, and further in view of Bansal et al., USPubN: 2003/0120593 (hereinafter Bansal)

As per claim 15, Labadie discloses a method for monitoring performance of at least two Java transactions that are related to one another as parent-child transactions, comprising obtaining a performance metric (e.g. *response time* - para 0005-0014, pg. 1-2) corresponding to each of said at least two Java transactions by:

installing an instrument hook prior to execution of each of said at least two Java transactions (e.g. Fig. 4A-C; *event correlator ...time stamp even for inclusion of ...a correlator* - para 0061, pg. 5; Fig. 5A – Note: metric gathering calls inserted within transaction threads via plug-in and ARM service implementation with provision of dynamic OO class and methods read on hooking 2 selected Java transactions within the control of the DCS system),

wherein a top level transaction of the at least two Java transactions is initiated in response to a request received from a web server (para 0032-0034, pg 3; Fig. 2 – Note: server receiving a inboud request from client DCS reads on initiating a start for a top level leading to detecting of more partner correlation or associated correlators with respect to said top level start request from DCS client) and

instrumenting each of said at least two Java transactions upon execution of each of said at least two Java transactions using one or more plug-in instruments called by the instrument hook (refer to claim 1);

generating a correlator corresponding to said parent transaction (para 0012-0013, pg. 2; Fig. 5B; *SOAP parameters, timestamp* – para 0073, pg. 7; Fig. 6A-C - Note: ARM with correlation service reads on corresponding correlator of child and that of parent);

generating another correlator corresponding to said child transaction (e.g. Fig. 5B; *SOAP parameters, timestamp* – para 0073, pg. 7; Fig. 6A-C - Note: ARM with correlation service reads on corresponding correlator of child and that of parent); and

and wherein the one or more plug-in instruments implement an interface that communicates data for the performance metric (e.g. class 310 *implements* an service provider – para 0059, pg 5; Fig. 2; *response time* - para 0005-0014, pg. 1-2; Fig. 5B; *SOAP parameters, timestamp* – para 0073, pg. 7; Fig. 6A-C; *plug-in* - para 0034-0035, pg. 4)

Labadie does not explicitly disclose wherein said web server transmits a cookie to an application server and utilizing said cookie to generate said correlator for said top level transaction. But the server's transmitted cookie being used to generate additional correlators for identifying children transactions associated with top level has been addressed in claim 1.

Labadie discloses utilizing RMI (see ORB, para 0028, pg. 3) to send said top-level correlator incorporated in a header of an IIOP message to said child transaction upon execution, and generating another correlator corresponding to said child transaction (Fig. 5A-B; *header* - para 0064-0066, pg. 6); but does not explicitly teach *RMI over IIOP*. The use of message over IIOP in a J2EE based network is taught by Bansal (Fig. 23) who also teaches using of ARM to instrument and measure application data for performance reporting (see para 0922-0969, pg. 38-40). Since Labadie is also suggesting performance analysis in a similar context where message containing correlator data are passed in a interoperability Enterprise Java network (para 0026, pg. 3), it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide a layer of IIOP as in Bansal's message passing above among the ORB layer pertinent to this J2EE paradigm in order for Labadie's RMI invocation (over ORB) or correlator

record passing to benefit of the core service of the ORB based on IIOP as heterogeneous format data can be reconverted from one format into another to fulfill the path of the data being transferred in this enterprise communications means.

As per claims 16-19, these claims include the subject matter of claims 3-5 or 6; hence will incorporate the corresponding rejection as set forth therein, respectively.

Response to Arguments

9. Applicant's arguments filed 6/17/08 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

Provisional Nonstatutory obviousness Double Patenting:

(A) Applicants have submitted that Labadie pursuant to MPEP § 804 is not applicable for a double patenting rejection (Appl. Rmrks,pg. 7). The nonstatutory section under (form § 8.37) which the double patenting is based upon addresses obviousness on conflicting claims between 2 co-pending applications, the instant application and case 10/640,619; that is, both of which are co-owned by same assignee *Hewlett-Packard Development Company*, and both including *Borkan Martha* as one inventor, and this fits § 804 in terms of "two or more ... applications ... must have at least one inventor and/or ... commonly assigned ...". Thus, form paragraph § 8.37 clearly establishes that conflicting claims between instant application and a first reference, case 10/640,619, such conflict being basis for the obviousness rationale. The first reference is co-pending case 10/640619 whose claimed subject matter conflicts with that of the instant application, whereas the second reference (as per its subject matter, not claims) Labadie is applied to support a case of obviousness. Labadie does not fall under the above MPEP 804 rule because, with respect to the instant application, Labadie is not a copending co-assigned

application; nor does Labadie have one common inventor, nor that Labadie's claim language conflict with the instant claimed subject matter. The argument is utterly misplaced.

USC § 103 Rejection:

(B) Applicants alleged that defining a method for distributed logging as proffered in the cited portions does not teach or suggest 'installing … hook prior to execution of the selected transaction' (Appl. Rmrks pg. 8, middle). The rejection has identified ARM methodology including instrumentation having source data, class structure based on which correlators, counter, and identifiers are defined to effectuate a instrumenting of transactional performance/behavior and threads spawned as a result of a runtime. The 'prior to' execution time is clearly taught. The argument is deemed insufficient factual rebuttal in order to overcome the above teachings by Labadie.

(C) Applicants have submitted that a provider plug-in defining a logging service does not teach or suggest (Appl. Rmrks pg. 9, top) that an 'instrument hook' calls a plug-in instrument in order to instrument a transaction (*). The 'plug-in instrument' can be interpreted as a instance of dynamically created API to implement a class purported a instrumentation service, such that such class invokes the creation or constructing of the API. The Tivoli service invokes a service of logging provided in form of plug-in DCS which in turn creates *interfaces* or API (see Labadie: *class 310, implements provider class* - para 0059, pg. 5) implementing a provider class in order to execute instrumentation such as logging of events by this DCS. This 'plug-in instrument' is construed as the dynamically created interface (or API) to implement this *provider class* of said DCS service; wherein the DCS class provider reads on a instrumentation type ARM or runtime hook invoking said plug-in class/interface; i.e. this provider class viewed as a instrumentation

hook or instrument object in view of the correlator needed within ARM-related API in Tivoli, regarding which the DCS classes acts a middleware object (e.g. for logging hook) between high level server and client data (see Labadie: para 0035, pg. 4; ARM API, para 0005, pg. 1); and this is deemed satisfying the above limitation (*).

(D) Applicants have submitted that the cited portions in Labadie does not teach 'initiating said top level transaction in response to … request from a web server' (Appl. Rmrks pg. 9, middle). The cited portions teach high level request flowing from a high server through middleware instantiation of interface then to the corresponding layer of the client side; and this is deemed initiating at the top in response to a request from a server. The claim is not provided with compelling details about this top-level transaction to preclude the hierarchy of data communication back and forth between the entities in Labadie's Figure 2 from reading onto the above quoted language. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference.

(E) Applicants have submitted that the language 'generating correlators for identifying said top level … and parent transaction' is not taught in Labadie. The hierarchy of parent child transaction has been shown in Labadie, and the ARM to utilize the needed correlator via the DCS invocation and instantiating of interfaces would be sufficient to disclose that correlator is for identifying a identification of a parent name or a child name, and this has been taught in the instrumentation portions and the very nature of correlating a parent transaction to its child transaction by ARM and Tivoli's API service(see Labadie: Fig. 3-4 for the DCS logging service; *parent transaction, child transaction* - ARM service – para 0005, pg 1), which Labadie's

correlator usage is all about, i.e. the methodology of using correlator teaches correlating live data within a transactional flow or entities in terms of a parent and child transaction, and subsequently among peer-to-peer layers of such hierarchy.

(F) Applicants have submitted that the cited portions by the Office Action fail to teach or suggest 'utilizing said correlators to cross-correlate a … metric … parent transaction … child transaction' (Appl. Rmrks pg. 10, middle) because only child transaction are identified. The correlator-based instrumentation is implemented to track transaction related parameters in the bi-directional flowing of events in the paradigm wherein parent transaction invokes a child transaction. Since Applicants fail to point where exactly in Labadie the instrumentation or ARM service forego identifying parent transaction and confine itself to only tracking child-and-child or peer-and-peer transactions, the broad interpretation of ARM and Tivoli cannot be treated as a mere child to child type of correlating as alleged in the argument, which would be teaching away from this well-known methodology; i.e. that ARM methodology is not Applicant's own work. Absent specifics in the claim as to preclude ARM used by Labadie from supporting a parent-child transaction type of instrumentation or correlation, the argument is deemed misplaced. The argument is referred to observations made in section D, with regard to the very nature of correlating in Tivoli and ARM methodology.

In all, the claims will stand rejected as set forth in the Office Action.

Conclusion

10. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571)272-3759.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.
Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tuan A Vu/

Primary Examiner, Art Unit 2193

September 05, 2008